
Vulnerability Assessment and Penetration Testing Report of
SmartCare Remote Management Web Application

Dec 2022

About L&T Technology Services:

L&T Technology Services Limited (LTTS) is a global leader in Engineering and R&D (ER&D) services. With 399 patents filed for 51 of the Global Top 100 ER&D spenders. Our innovations speak for itself – World's 1st Autonomous Welding Robot, Solar 'Connectivity' Drone, and the Smartest Campus in the World, to name a few. LTTS expertise in engineering design, product development, smart manufacturing, and digitalization touches every area of our lives. With 49 Innovation and R&D design centres globally, we specialize in disruptive technology spaces such as 5G, Artificial Intelligence, Collaborative Robots, Digital Factory, and Autonomous Transport.

LTTS is a publicly listed subsidiary of Larsen & Toubro Limited, the \$18 billion Indian conglomerate operating in over 30 countries.

Document History:

Rel. No.	Release Date	Prepared By. Prepared. Dt.	Reviewed By Reviewed Dt.	Approved By Approval Dt.	Remark/Revision Details
1.0	Dec 2022	Sanidya Sharan Sai Praneetha Bhaskaruni	Sunil M.C	Atanu Niyogi	Initial Version
		14 th Dec 2022			
2.0	Dec 2022	Sunil M C	Atanu Niyogi	Atanu Niyogi	Review comments
		23 rd Dec 2022	26 th Dec 2022		

Table of Contents

1. Overview of the project	5
2. Vulnerabilities explained in detail	7
2.1 Insecure Direct Object References (IDOR)	7
2.2 No Account Lockout Policy	9
2.3 Weak Password Policy	10
2.4 Improper Cookies Validation	11
2.5 Use of dangerous HTTP methods	12
2.6 No Content Security Policy	14
2.7 Encoding of credentials	15
2.8 Long Session Timeout	16
2.9 Lack of rate limit leads to Brute-Force attack	19
2.10 Sensitive Information disclosure	20
2.11 No Multifactor Authentication (MFA)	21
3. Abbreviation	23

1. Overview of the project

L&T Technology Services (LTTS) security team has conducted Security Assessment for the SmartCare Remote Management web application. The purpose of the assessment is to evaluate the security posture of the web application against common vulnerabilities.

Objective of the security assessment:

As a part of this engagement, a holistic approach was taken to conduct the Vulnerability Assessment and Penetration Testing on the SmartCare Remote Management web application. During the engagement High, Medium, and Low severity issues were identified with respect to the SmartCare Remote Management web application.

Approach

The following approach was taken to make sure the target was assessed against known vulnerabilities from all possible security perspectives:

- Manual Vulnerability Assessment and Penetration Testing using OWASP TOP 10 for web applications.

Some of the tools which were used are listed below:

Target Application	SmartCare Remote Management web application
Browser	Chrome, Firefox
Tools	Burp Suite, Nmap, Nikto, Whatweb, SQLMap, dirbuster

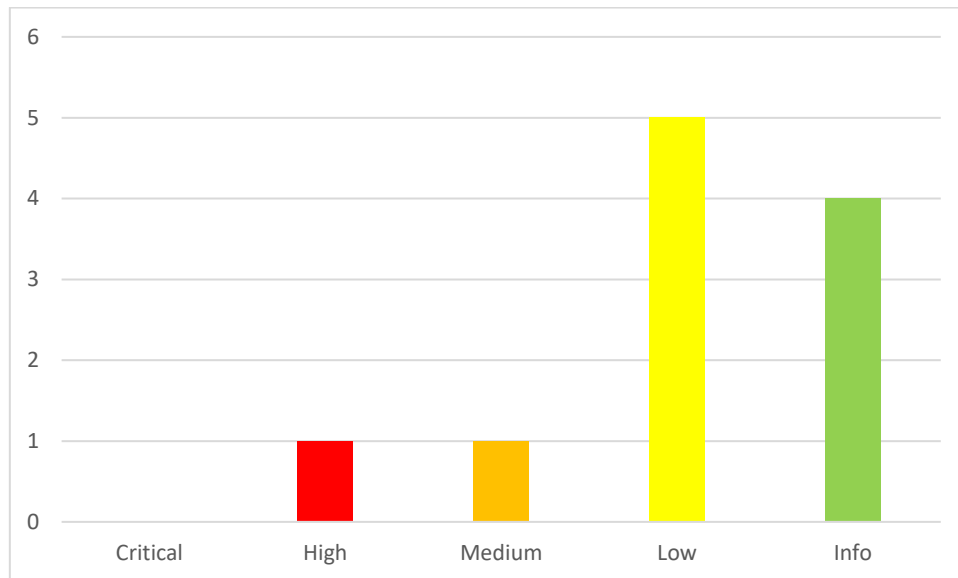
Key Security Policies

OWASP top 10 listed vulnerabilities were used as a reference framework. The following key security aspects were checked:

1. Broken Access Control
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery

Summary of Findings

The graph below shows a summary of the number of vulnerabilities found for each impact level for the Application Security Assessment. Vulnerabilities found are addressed according to priority, findings, analysis, and recommendations from the assessment.



Sr.no	Title	Risk Rating
1	Insecure Direct Object References (IDOR)	High
2	No Account Lockout Policy	Medium
3	Weak Password Policy	Low
4	Improper Cookies Validation	Low
5	Use of dangerous HTTP methods	Low
6	No Content Security Policy	Low
7	Encoding of Credentials	Low
8	Long Session Timeout	Info
9	Lack of rate limit leads to Brute-Force attack	Info
10	Sensitive Information Disclosure	Info
11	No Multifactor Authentication (MFA)	Info

Overall Risk Severity = Likelihood x Impact				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
Likelihood				

2. Vulnerabilities explained in detail

2.1 Insecure Direct Object References (IDOR)			
Impact	High	Risk Rating	High
Ease of Exploit	Easy		
Likelihood	Medium		
Category	Authorization Bypass Through User-Controlled Key		
URL/Impacted system	https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#		
Description			
<p>Insecure direct object references (IDOR) are a cybersecurity issue that occurs when a web application developer uses an identifier for direct access to an internal implementation object but provides no additional access control and/or authorization checks. For example, an IDOR vulnerability would happen if the curl of a transaction could be changed through client-side user input to show unauthorized data of another transaction.</p> <p>IDOR can also help attacker bypass RBAC (Role Based Access Control) where users who may not have permission to view other records can simply manipulate the parameters and get access. IDOR can result in vertical and horizontal privilege escalation scenarios.</p> <ul style="list-style-type: none">- https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/04-Testing for Insecure Direct Object References- https://portswigger.net/web-security/access-control/idor			
Impact			
Exposure of Confidential Information: When the attacker will have control over your account via this vulnerability, it is obvious that an attacker will be able to come across your personal information.			
Recommendation			
<ul style="list-style-type: none">• Developers should avoid displaying private object references such as keys or file names.• Validation of Parameters should be properly implemented.• Verification of all the Referenced objects should be done.• Tokens should be generated in such a way that they should only be mapped to the user and should not be public.			
How to recreate the Security defect			
<ul style="list-style-type: none">• Login to the web application.• Capture the user profile request with the burp suite.• Forward the request to an intruder.• Give payloads (100-999 numbers) in the devices Id position• After a successful attack, we can get other device’s details.			
Evidence			

Security Assessment for SCRM

Positions

Choose an attack type

Attack type: Sniper

Payload Positions

Configure the positions where payloads will be inserted, they can be added into the target

Target: https://scrmcf.sandboxiot.hillrom.com

```
1 GET /inventory/managedObjects/237611 HTTP/1.1
2 Host: scrmcf.sandboxiot.hillrom.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:108.0)
4 Accept: application/json
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: https://scrmcf.sandboxiot.hillrom.com/apps/remotemanagement/index.html
8 Authorization: Basic dDI0NzRkNjAveGVuX3Rlc3RAdXNlcj5hCj06VGZdEAXMjM=
9 X-Cumulocity-Application-Key: smartcare-application-key
10 User-Agent: true
11 Sec-Fetch-Dest: empty
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Site: same-origin
14 Te: trailers
15 Connection: close
16
17
```

1 payload position

Results

Positions

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Cookies	Comment
509	608	404			585		
510	609	404			585		
511	610	404			585		
512	611	200			3007		
513	612	404			585		
514	613	404			585		

Request

Response

Pretty

Raw

Hex

```
1 GET /inventory/managedObjects/237611 HTTP/1.1
2 Host: scrmcf.sandboxiot.hillrom.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:108.0) Gecko/20100101 Firefox/108.0
4 Accept: application/json
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: https://scrmcf.sandboxiot.hillrom.com/apps/remotemanagement/index.html
8 Authorization: Basic dDI0NzRkNjAveGVuX3Rlc3RAdXNlcj5hCj06VGZdEAXMjM=
9 X-Cumulocity-Application-Key: smartcare-application-key
10 User-Agent: true
11 Sec-Fetch-Dest: empty
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Site: same-origin
14 Te: trailers
15 Connection: close
16
17
```

0 matches

Clear

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Cookies	Comment
509	608	404			585		
590	689	404			585		
591	690	200			1490		
592	691	404			585		
593	692	404			585		

Request

Response

Pretty

Raw

Hex

```
1 GET /inventory/managedObjects/237690 HTTP/1.1
2 Host: scrmcf.sandboxiot.hillrom.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:108.0) Gecko/20100101 Firefox/108.0
4 Accept: application/json
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: https://scrmcf.sandboxiot.hillrom.com/apps/remotemanagement/index.html
8 Authorization: Basic dDI0NzRkNjAveGVuX3Rlc3RAdXNlcj5hCj06VGZdEAXMjM=
9 X-Cumulocity-Application-Key: smartcare-application-key
10 User-Agent: true
11 Sec-Fetch-Dest: empty
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Site: same-origin
14 Te: trailers
15 Connection: close
16
17
```

0 matches

Clear

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Cookies	Comment
589	688	404			585		
590	689	404			585		
591	690	200			1490		
592	691	404			585		
593	692	404			585		

Request

Response

Pretty

Raw

Hex

Render

```
1 HTTP/1.1 200 OK
2 Date: Fri, 16 Dec 2022 09:46:35 GMT
3 Content-Type: application/vnd.com.nsn.cumulocity.managedobject+json;charset=UTF-8;ver=0.9
4 Content-Length: 1124
5 Connection: close
6 Cache-Control: no-cache,no-store,must-revalidate
7 Pragma: no-cache
8 Expires: -1
9 X-Content-Type-Options: nosniff
10 Strict-Transport-Security: max-age=31536000; includeSubDomains
11
12 {
  "additionParents": {
    "references": [
      ],
      "self": "https://t2479160.sandboxiot.hillrom.com/inventory/managedObjects/237690/additionParents"
    },
    "owner": "kenneth.gilbert@baxter.com",
    "childDevices": {
      "references": [
      ],
      "self": "https://t2479160.sandboxiot.hillrom.com/inventory/managedObjects/237690/childDevices"
    },
    "childAssets": {
      "references": [
      ],
      "self": "https://t2479160.sandboxiot.hillrom.com/inventory/managedObjects/237690/childAssets"
    },
    "creationTime": "2022-09-22T19:52:45.149Z",
  }
}
```

0 matches

Clear

Security Assessment for SCRM

The screenshot shows the Burp Suite interface. On the left, the 'Positions' tab is active, showing a list of positions. The 'Results' tab on the right displays a table with columns: Request, Payload, Status, Error, Timeout, Length, Cookies, and Comment. The response is a JSON object containing 'additionParents' and 'references'.

2.2 No Account Lockout Policy

Impact	Medium	Risk Rating	Medium
Ease of Exploit	Easy		
Likelihood	Medium		
Category	Broken Authentication		
URL/Impacted system	https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#/		

Description

During Assessment, we observed SCRM web application doesn't follow the account lockout policy. Current mechanism allows for n number of attempts to occur on any username. This gives attackers the bandwidth needed to carry out brute force attacks. If the attackers know the username, then the brute force on password field can happen with increased probability. If the attackers do not know the username, then they can still configure utilities to run dictionary based or other variants to run on SCRM for longer durations.

Impact

It is possible for an attacker to gain access to the application by brute-forcing the password. Since there are no restrictions on the number of logins attempts a malicious user can brute force the credentials of a user until the right credentials are guessed. The impact is limited for SCRM as the project is almost migrated to a SSO based mechanism.

Recommendation

While SCRM may adopt a SSO based authentication mechanism. We recommend having a account lockout mechanism still in place if there are a minor group of user who may still use a username/password based authentication method. However, if all users are migrated to a Single Sign-On mechanism then the vulnerability can be considered irrelevant.

How to recreate the Security defect

- Login into the SCRM web application.
- Configure Burp or any other utility (like Zap) to continuously execute login payload with input coming from a dictionary file.

- We observed that login into the account without any logout.

Evidence

The screenshot shows the Burp Suite interface. On the left, the 'Positions' tab is active, showing a list of positions. The 'Payload Positions' section is expanded, showing a list of positions. The 'Target' is set to <https://scrmcf.sandboxiot.hillrom.com>. The 'Request' tab is selected, showing a list of requests. The selected request is a GET to https://scrmcf.sandboxiot.hillrom.com/tenant/currentTenant_HTTP/1.1 with a Basic authentication header. The response is a 401 Unauthorized status.

2.3 Weak Password Policy

Impact	Low	Risk Rating	Low
Ease of Exploit	Moderate		
Likelihood	Medium		
Category	Weak Password Requirements		
URL/Impacted system	https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#/		

Description

During the assessment of the application, it was observed that the application's authenticated password reset mechanism uses the existing password or current password to execute a password change. Accepting the previous password is a bad policy. The reason of change the passwords is security, if the old password still works there's no point in changing it. So, make sure that the old password is useless.

The severity of the vulnerability is set to low as SCRM will be using SSO based mechanism for authentication and authorization which will shift the responsibility of the password management to the Identity and Access Management platform. However, the vulnerability is still recorded as reference so that the test scenario can be assessed with the SSO based mechanism as well.

Impact

- If the old password was revealed to someone else (which can happen, especially with phishing) then there is a chance to login with the credentials and can gain access to the application.

- It is possible someone may gain access to your saved passwords.

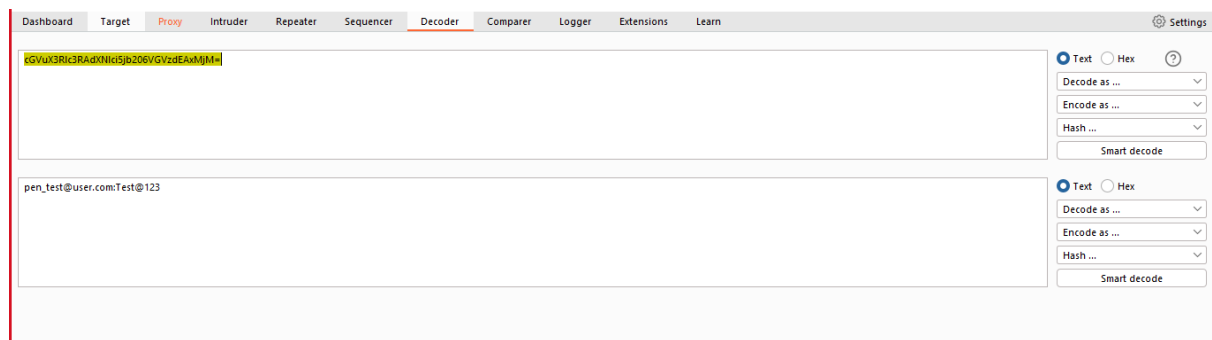
Recommendation

- Changing a user's password after a fixed amount of time is requested to keep the account reasonably safe if the authentication credentials are stolen or leaked. Hence, accepting previous passwords is an unsafe solution and should be avoided.
- Introduce additional authentication controls (i.e., multi-factor authentication).
- Set minimum password length to at least a value of 12. If the number of characters is set to 0, no password is required. In most environments, an eight-character password is recommended because it's long enough to provide adequate security
- Also recommended is to test the password based scenarios once the SSO mechanism is in place and report if there are any vulnerabilities to the Identity and Access Management (IAM) team.

How to recreate the Security defect

- Browse to - <https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#/>
- Capture the request in burp and decode the authorization header.
- We observed that the application doesn't follow the password policy.
- Notice that the application accepts the old password.

Evidence



2.4 Improper Cookies Validation

Impact	Medium	Risk Rating	Low
Ease of Exploit	Easy		
Likelihood	Low		
Category	Reliance on Cookies without Validation and Integrity Checking		
URL/Impacted system	<u>https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#</u> <u>/</u>		
Description			
Random value set Cookies no server checks in place to authenticate the ID.			
Impact			

Authentication bypass vulnerability could allow attackers to perform various malicious operations by bypassing the device authentication mechanism. If a hacker can get access to valid cookies, then they can carry out authentication bypassing by doing a session hijack attack – essentially providing the server with the Cookies of someone who has already been authenticated, thereby impersonating them.

Recommendation

- They suggest ensuring that the user cookie values are set with a unique value after the user has successfully authenticated.
- The cookie values must be with sufficient length and randomized values as per (https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html)
- It is best to have a secure and strong authentication policy in place.
- It is best to ensure all systems, folders, and apps are password protected.
- It is suggested to not expose authentication protocol in the client-side web browser script.

How to recreate the Security defect

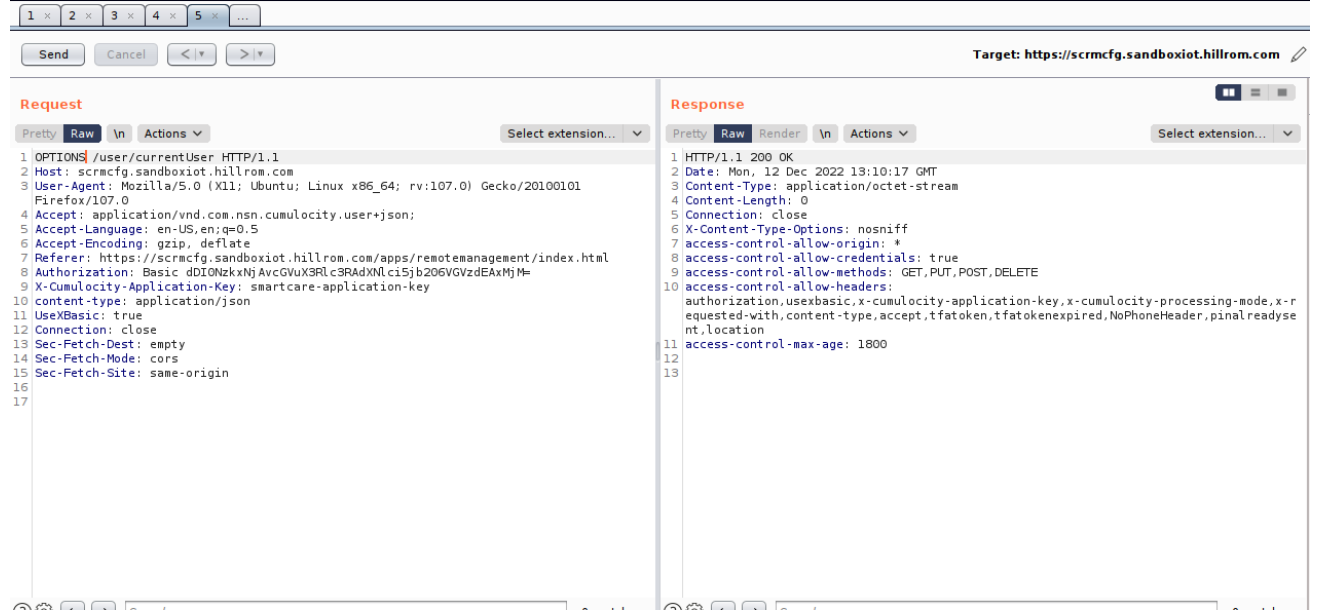
- Login to the user account.
- Capture the request in Burp Suite.
- Manipulate the Cookies ID and refresh the Page.
- It was observed, it is working with manipulated Cookies ID.

Evidence

The screenshot shows the Burp Suite interface with a target URL of `https://scrmcfg.sandboxiot.hillrom.com`. The left pane displays the 'Request' tab for a GET request to `/tenant/currentTenant`. The right pane displays the 'Response' tab, showing a 200 OK status and JSON data. A red box highlights the cookie `sk_bscPentest123; AKA_A2=A` in the request headers.

2.5 Use of dangerous HTTP methods

Impact	Medium	Risk Rating	Low
Ease of Exploit	Moderate		
Likelihood	Low		
Category	CWE: 650 Trusting HTTP Permission Methods on the Server Side		

URL/Impacted system	https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#/
Description	
<p>During Pentest, we observed SCRM web application doesn't follow the OPTIONS method and provides a list of the methods that are supported by the web server, it represents a request for information about the communication options available on the request/response chain identified by the Request-URI. HTTP offers several methods that can be used to perform actions on the web server. Many of these methods are designed to aid developers in deploying and testing HTTP applications. These HTTP methods can be used for nefarious purposes if the web server is misconfigured. Additionally, Cross Site Tracing (XST), a form of cross-site scripting using the server's HTTP TRACE method, is examined. While GET and POST are by far the most common methods that are used to access information provided by a web server, the Hypertext Transfer Protocol (HTTP) allows several other (and somewhat less known) methods.</p>	
Impact	
<p>The OPTIONS method may expose sensitive information that may help a malicious user to prepare more advanced attacks such as:</p> <p>OPTIONS DELETE HEAD PATCH</p>	
Recommendation	
Options, Put, Delete, and Head methods should be disabled.	
How to recreate the Security defect	
<ul style="list-style-type: none"> • Login into the SmartCare Remote Management web application. • Capture the traffic into burp and Observe OPTIONS METHOD • We can use all the HTTP Methods shown. 	
Evidence	
 <p>The screenshot shows a Burp Suite interface with a target URL of https://scrmcfg.sandboxiot.hillrom.com. On the left, the 'Request' tab is active, displaying an OPTIONS request. The request body is: <code>OPTIONS /user/currentUser HTTP/1.1</code>. The request headers include Host, User-Agent, Accept, Accept-Language, Accept-Encoding, Referer, Authorization, X-Cumulocity-Application-Key, Content-Type, and User-Agent. On the right, the 'Response' tab is active, displaying an HTTP 200 OK response. The response headers include Date, Content-Type, Content-Length, Connection, X-Content-Type-Options, access-control-allow-origin, access-control-allow-credentials, access-control-allow-methods, access-control-allow-headers, and access-control-max-age. The access-control-allow-methods header lists: GET, PUT, POST, DELETE.</p>	

2.6 No Content Security Policy

Impact	Low	Risk Rating	Low
Ease of Exploit	Medium		
Likelihood	Medium		
Category	Improper Restriction of Rendered UI Layers or Frames		
URL/Impacted system	https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#/		

Description

During the HTTP traffic analysis of the SCRM Web interface, it was observed that the server does not support Content Security Policy. Content Security Policy is a standard that helps protect against various content injection attacks like cross-site scripting. While the victim is interacting with seemingly harmless web pages.

Impact

Without a Content Security Policy, an attacker can perform content injection attacks if data from the service is displayed in a browser

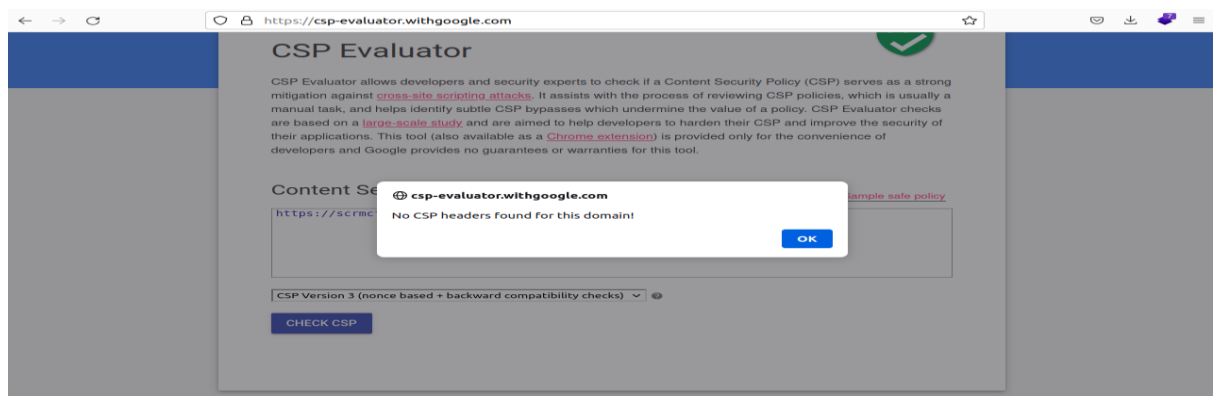
Recommendation

Enabling the Content Security Policy response header to all HTTP server responses helps in preventing content injection attacks. While adding Content Security Policy it must be set correctly specifying the locations from which content can be loaded. Content-Security-Policy: <Policy-directive>;

How to recreate the Security defect

- Browse to - <https://csp-evaluator.withgoogle.com/>
- Enter the URL -
- Click on check CSP

Evidence



2.7 Encoding of credentials			
Impact	Medium	Risk Rating	Low
Ease of Exploit	Moderate		
Likelihood	Medium		
Category	CWE-261: Weak encoding for password		
URL/Impacted system	https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#/		
Description			
<p>Base 64 is NOT a valid encryption method for passwords if you want it to be secure. It is barely better than plaintext.</p> <p>Usually, the password is hashed and salted, and then checking passwords involves hashing the new string with the same method and checking against the stored hash.</p> <p>While encoding the username and password with the Base64 algorithm typically makes them unreadable by the naked eye, they are as easily decoded as they are encoded. Security is not the intent of the encoding step.</p>			
Impact			
<p>Weak encryption can be decoded easily. Hence can result in sensitive data exposure, key leakage, broken authentication, insecure session, and spoofing attacks.</p> <p>Using Basic authorization can lead to replay attacks.</p> <p>In HTTP basic authentication, the client receives an authentication token from the server, which is constructed by concatenating the username and password, and encoding it in Base64. This token is stored and managed by the browser, which automatically adds it to the Authorization header of every subsequent request as follows:</p> <p><i>Authorization: Basic base64(username:password)</i></p> <p>For several reasons, this is generally not considered a secure authentication method. Firstly, it involves repeatedly sending the user's login credentials with every request. Unless the website also implements HSTS, user credentials are open to being captured in a man-in-the-middle attack.</p> <p>In addition, implementations of HTTP basic authentication often don't support brute-force protection. As the token consists exclusively of static values, this can leave it vulnerable to being brute forced.</p> <p>HTTP basic authentication is also particularly vulnerable to session-related exploits, notably CSRF, against which it offers no protection on its own.</p> <p>In some cases, exploiting vulnerable HTTP basic authentication might only grant an attacker access to a seemingly uninteresting page. However, in addition to providing a further attack surface, the credentials exposed in this way might be reused in other, more confidential contexts.</p>			
Recommendation			
Use strong encryption algorithms like AES, DES, Triple DES.			
How to recreate the Security defect			

- Login to the web application–
<https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#/>
- Capture the traffic using burp
- Look for Authorization: Basic in the requests which has parameters as shown in the below evidence.

Evidence

The screenshot displays the Burp Suite interface. At the top, a request to `https://scrmcfg.sandboxiot.hillrom.com:443 [20.81.62.223]` is shown with buttons for `Forward`, `Drop`, `Intercept is on`, `Action`, and `Open Browser`. Below this, the request is shown in `Raw` format. The request details are as follows:

```

1 GET /tenant/currentTenant HTTP/1.1
2 Host: scrmcfg.sandboxiot.hillrom.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:108.0) Gecko/20100101 Firefox/108.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html
8 Authorization: Basic eGVuX3Rlc3RAdXNlcj5jb206VGZdEAxMjM=
9 X-Cumulocity-Application-Key: smartcare-application-key
10 Content-Type: application/json
11 User-Agent: true
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15 Te: trailers
16 Connection: close
17
18

```

The `Authorization: Basic eGVuX3Rlc3RAdXNlcj5jb206VGZdEAxMjM=` header is highlighted with a red box. Below the request, the `Decoder` tab is active, showing the decoded content of the `Authorization` header. The decoded content is `pen_test@user.com:Test@123`.

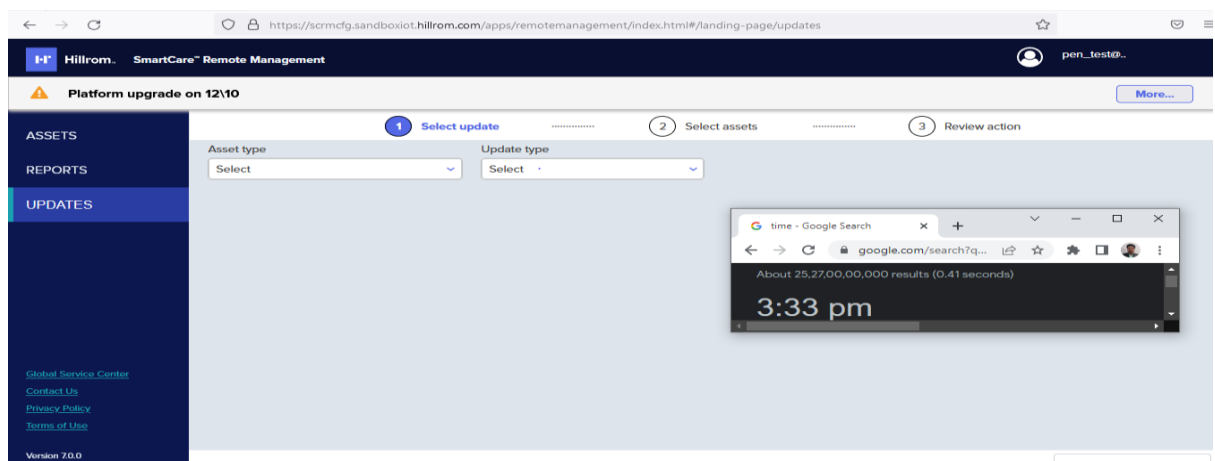
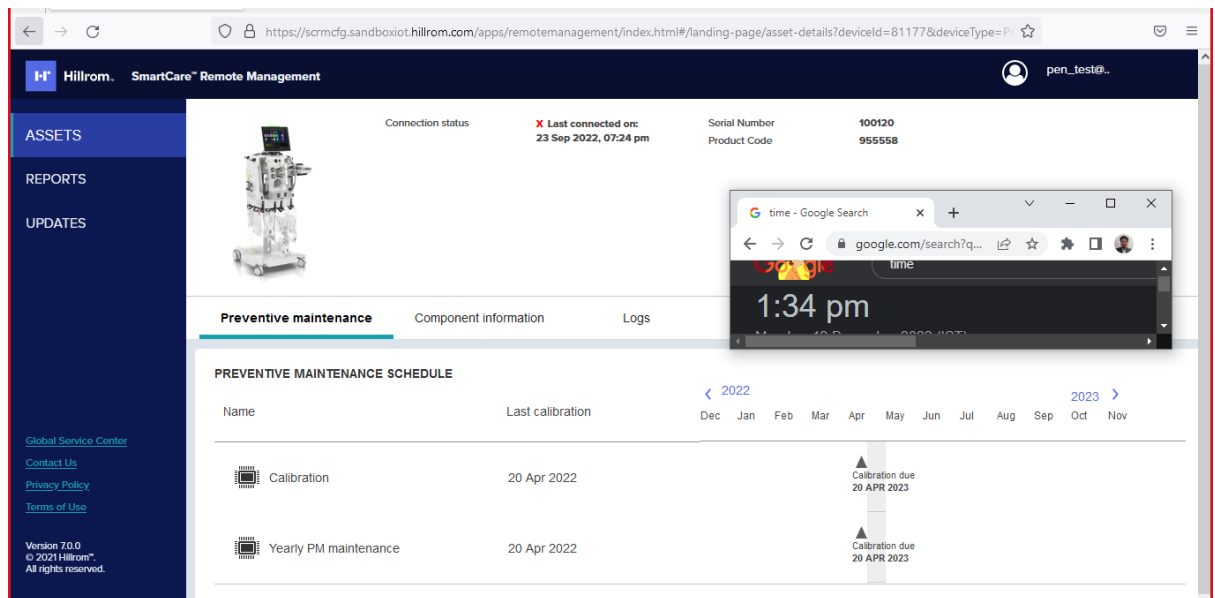
2.8 Long Session Timeout

Impact	Low	Info
--------	-----	------

Ease of Exploit	Easy	Risk Rating	
Likelihood	Low		
Category	Session Management		
URL/Impacted system	https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#/		
Description			
During the assessment, we observed an attacker's ability to hijack a victim's session increases proportionally with the amount of time an idle user's session remains valid. Once a valid session identifier is obtained, the attacker can impersonate the victim in the application, performing any functionality and accessing any data made available to the victim. Currently for SCRM the session timeout is set at 120 min.			
Impact			
<ul style="list-style-type: none">Insufficient Session timeout increases a Web site exposure to attacks that steal or reuse user's session identifiers.Cookie Hijacking is possible; application integrity can be compromised.Once a valid session identifier is obtained, the attacker can impersonate the victim in the application.			
Recommendation			
Terminate the user's session server-side after a sufficiently short idle period. When the user makes further requests using the expired session, they should be redirected to a splash page or the login pages. In addition, the client-side code should track session idle time and automatically redirect the user to a splash page or the login page after a certain period of client-side inactivity has passed. No prior authenticated user data or functionality should continue to be displayed after the timeout occurs. Determine a session timeout duration that sufficiently protects end users and the application while maintaining system usability. Session timeouts of 30-120 minutes are common for most web applications and vary depending on the sensitivity of the information available during each session. Various standards organizations and government entities also typically recommend organization-defined timeouts or call for an idle timeout of 30-120 minutes: <ul style="list-style-type: none">PCI DSS v3.2 section 8.1.8 states, "If a session has been idle for more than 15 minutes, require the user to re-authenticate to re-activate the terminal or session."U.S. CNSS - CNSSI No. 1253 section AC-11 states, "Session Lock ... not to exceed 30 minutes"NIST SP800-53 section AC-11 states, "...Prevents further access to the system by initiating a session lock after [Assignment: the organization-defined period of inactivity or upon receiving a request from a user".			
How to recreate the Security defect			

- Login to the application
- Idle the application for up to 30-60 minutes
- Can access the application even after 30-60 minutes

Evidence



2.9 Lack of rate limit leads to DOS Attack

Impact	Low	Risk Rating	Info
Ease of Exploit	Moderate		
Likelihood	Low		
Category	Broken Access Control		
URL/Impacted system	https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#/		

Description

During the penetration testing, we observed a lack of a rate limit on the login page. This vulnerability leads to brute force when an attacker tries to brute-force of username and password on the login page. On the login page, the attacker tries to brute-force the user credentials. When a user wants to reset his password and there is no rate limiting on the function, an attacker can take this as an advantage.

Impact

This vulnerability leads to a DOS attack and the attacker can perform brute force attacks also using a username and password.

The impact for this vulnerability with respect to SCRM is very minimal as SCRM intends to use a SSO based authentication mechanism.

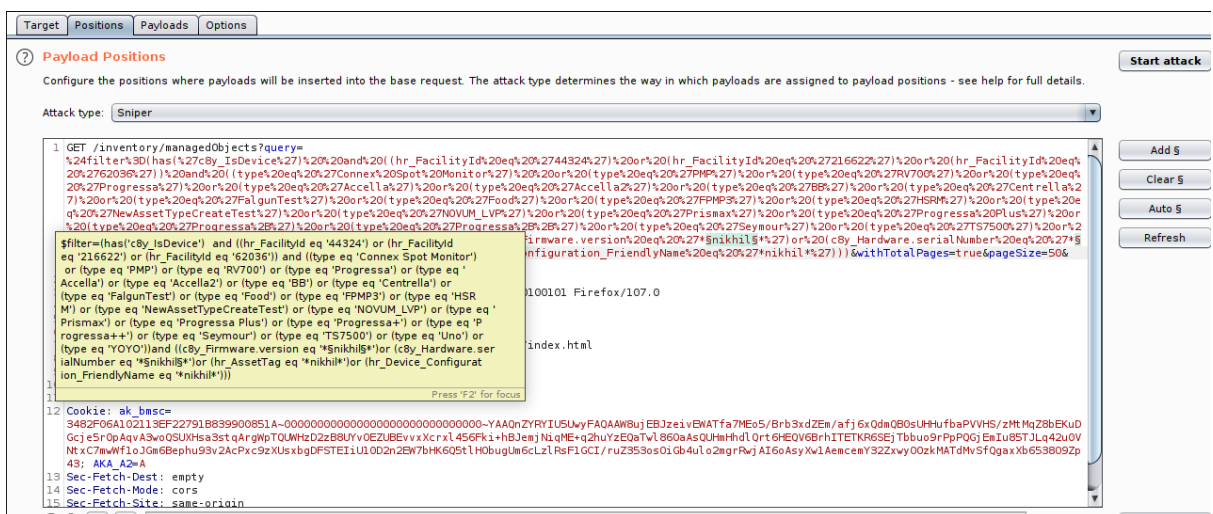
Recommendation

To mitigate this issue developers should implement a timeout after several requests in a period or implement a CAPTCHA mechanism on the form page.

How to recreate the Security defect

- Tried to login in as a user account.
- Capture the traffic into the burp suite and send it to the intruder tab.
- Give 150 Payload on Password input.
- It was observed no error was found till all payloads execute.

Evidence



Results	Target	Positions	Payloads	Options			
Filter: Showing all items							
Request	Position	Payload	Status	Error	Timeout	Length	Comment
362	1	<img src/x32=x onerror="... 200	200			1712	
360	1	<img src/x10=x onerror="... 200	200			1712	
361	1	<img src/x3=x onerror="... 200	200			1712	
359	1	<img src/x09=x onerror="... 200	200			1712	
364	1	<img src/x11=x onerror="... 200	200			1712	
367	1	<img src=xv09onerror="j... 200	200			1709	
365	1	<img src/x00=x onerror="... 200	200			1712	
366	1	<img src/x47=x onerror="... 200	200			1712	
368	1	<img src=x/x10onerror="j... 200	200			1709	
372	1	<img[a b c]src[d]=x[e]o... 200	200			1731	
371	1	<img src=x/x13onerror="j... 200	200			1709	
373	1	<img src=x onerror="x09"... 200	200			1712	
369	1	<img src=x/x11onerror="j... 200	200			1709	
370	1	<img src=x/x12onerror="j... 200	200			1709	
376	1	<img src=x onerror="x12"... 200	200			1712	
377	1	<img src=x onerror="x32"... 200	200			1712	

Request	Response
Pretty	Raw in Actions
<div> <div>Select extension...</div> <pre> 1 GET /inventory/managedobjects?query= %24filter%3D(has%27cby_IsDevice%27)%20%20and%20((hr_FacilityId%20eq%20%2744342%27)%20or%20(hr_FacilityId%20eq%20%2721662%27)%20or%20(hr_FacilityId%20eq%20%2762036%27))%20and%20(((type%20eq%20%27Connex%20Spots%20Monitor%27)%20%20or%20(type%20eq%20%27PMP%27)%20or%20(type%20eq%20%27FV700%27)%20or%20(type%20eq%20%27Progress%27)%20or%20(type%20eq%20%27Accella%27)%20or%20(type%20eq%20%27Accell%27)%20or%20(type%20eq%20%27BBS%27)%20or%20(type%20eq%20%27Centrella%27)%20or%20(type%20eq%20%27FalgunTest%27)%20or%20(type%20eq%20%27Food%27)%20or%20(type%20eq%20%27FPH%27)%20or%20(type%20eq%20%27HSRM%27)%20or%20(type%20eq%20%27NewAssetTypeCreatTest%27)%20or%20(type%20eq%20%27NOVUM_LVP%27)%20or%20(type%20eq%20%27Prisma%27)%20or%20(type%20eq%20%27Progress%20Plus%27)%20or%20(type%20eq%20%27Progress%27)%20or%20(type%20eq%20%27Progress%27%27)%20or%20(type%20eq%20%27Progress%27%27)%20or%20(type%20eq%20%27Seymour%27)%20or%20(type%20eq%20%27T75500%27)%20or%20(type%20eq%20%27Unot%27)%20or%20(type%20eq%20%27YOY%27))and%20(cby_Firmware.version%20eq%20%27%3cing%20src%5c3d%20oneror%3d%22)avascr ipt%3Aalert(1)%22%3e%27)or%20(cby_Hardware.serialNumber%20eq%20%27%nikhil%27)or%20(hr_AssetTag%20eq%20%27%nikhil%27)or%20(hr_Device_Configuration_FriendlyName%20eq%20%27%nikhil%27))&withTotalPagesetSize=50&currentPage=1 HTTP/1.1 2 Host: scrmcfc.sandboxiot.hillrom.com 3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:107.0) Gecko/20100101 Firefox/107.0 4 Accept: application/json 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate 7 Referer: https://scrmcfc.sandboxiot.hillrom.com/apps/remotemanagement/index.html 8 Authorization: Basic dDlONzkxMjAvGvcXSRlc3RAdXNlcj5jB26GVGVzZAEaXtMjM= </pre> </div>	

2.10 Sensitive Information disclosure

Impact	Low	Risk Rating	Info
Ease of Exploit	Moderate		
Likelihood	Medium		
Category	Exposure of Resource to Wrong Sphere		
URL/Impacted system	https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#/		

Description

During the penetration testing, we observed that email addresses is disclosed in the application response.

Impact

This can lead to Brute-force attacks. Attacker can use the sensitive information disclosed of account with higher privileges and functionality misuse or compromise account.

Another impact of revealing PII is it leads to phishing or spear phishing attacks. Attackers can target the email id obtained for such spear phishing scenarios and it may lead to malicious payloads sent to victims inbox.

Recommendation

- Mask the PII data if it is not used in the frontend of the application.
- If not needed from the UI layer then try to restrict the data in the response.

How to recreate the Security defect

- Login to the web application–
<https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#/>
- Capture the traffic using burp
- Check the application response containing the email id.

Evidence

Send Cancel < > Target: <https://scrmfg.sandboxiot.hillrom.com> HTTP/1

Request

```

1 GET /inventory/managedObjects/279203 HTTP/1.1
2 Host: scrmfg.sandboxiot.hillrom.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:107.0) Gecko/20100101 Firefox/107.0
4 Accept: application/json
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: https://scrmfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html
8 Authorization: Basic dDl0HkxNjAvcGVuO3Rlc3RAdQhlcj5jb206VGVzdEAnNjM=
9 X-Cumulocity-Application-Key: smartcare-application-key
10 UserBasic: true
11 Sec-Fetch-Dest: empty
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Site: same-origin
14 Te: trailers
15 Connection: close
16
17

```

Response

```

1 Content-Type: application/vnd.com.nsm.cumulocity.managedobject+json;charset=UTF-8;ver=0.9
2 Content-Length: 2553
3 Connection: close
4 Cache-Control: no-cache,no-store,must-revalidate
5 Pragma: no-cache
6 Expires: -1
7 X-Content-Type-Options: nosniff
8 Strict-Transport-Security: max-age=31536000; includeSubDomains
9
10 {
11   "additionParents": {
12     "references": [
13       ],
14     "self":
15       "https://t2479160.sandboxiot.hillrom.com/inventory/managedObjects/279203/additionParents"
16     },
17     "owner": "poornima.kumar@hillrom.com",
18     "childDevices": {
19       "references": [
20         ],
21       "self":
22         "https://t2479160.sandboxiot.hillrom.com/inventory/managedObjects/279203/childDevices"
23       },
24     "childAssets": {
25       "references": [
26         ],
27       "self":
28         "https://t2479160.sandboxiot.hillrom.com/inventory/managedObjects/279203/childAssets"
29       },
30     "creationTime": "2022-10-11T03:10:12.916Z",
31     "type": "Progressa Plus",
32     "lastUpdated": "2022-12-08T14:50:56.534Z",
33   }
34 }

```

Send Cancel < > Target: <https://scrmfg.sandboxiot.hillrom.com> HTTP/1

Request

```

1 GET /inventory/managedObjects/279203 HTTP/1.1
2 Host: scrmfg.sandboxiot.hillrom.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:107.0) Gecko/20100101 Firefox/107.0
4 Accept: application/json
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: https://scrmfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html
8 Authorization: Basic dDl0HkxNjAvcGVuO3Rlc3RAdQhlcj5jb206VGVzdEAnNjM=
9 X-Cumulocity-Application-Key: smartcare-application-key
10 UserBasic: true
11 Sec-Fetch-Dest: empty
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Site: same-origin
14 Te: trailers
15 Connection: close
16
17

```

Response

```

1 Content-Type: application/vnd.com.nsm.cumulocity.managedobject+json;charset=UTF-8;ver=0.9
2 Content-Length: 2553
3 Connection: close
4 Cache-Control: no-cache,no-store,must-revalidate
5 Pragma: no-cache
6 Expires: -1
7 X-Content-Type-Options: nosniff
8 Strict-Transport-Security: max-age=31536000; includeSubDomains
9
10 {
11   "additionParents": {
12     "references": [
13       ],
14     "self":
15       "https://t2479160.sandboxiot.hillrom.com/inventory/managedObjects/279203/additionParents"
16     },
17     "owner": "poornima.kumar@hillrom.com",
18     "childDevices": {
19       "references": [
20         ],
21       "self":
22         "https://t2479160.sandboxiot.hillrom.com/inventory/managedObjects/279203/childDevices"
23       },
24     "childAssets": {
25       "references": [
26         ],
27       "self":
28         "https://t2479160.sandboxiot.hillrom.com/inventory/managedObjects/279203/childAssets"
29       },
30     "creationTime": "2022-10-11T03:10:12.916Z",
31     "type": "Progressa Plus",
32     "lastUpdated": "2022-12-08T14:50:56.534Z",
33   }
34 }

```

https://t2479160.sandboxiot.hillrom.com/inventory/managedObjects/279203/additionParents",

"owner": "pavithra.thanigachalam@softwareag.com",

"childDevices": {

"references": [

],

"self":

"https://t2479160.sandboxiot.hillrom.com/inventory/managedObjects/279203/additionParents"

},

"owner": "Nagesh. Rao@softwareag.com",

"childDevices": {

"references": [

],

"self":

2.11 No Multifactor Authentication (MFA)

Impact

Low

Info

Ease of Exploit	Easy	Risk Rating	
Likelihood	Low		
Category	Use of Single-factor Authentication		
URL/Impacted system	https://scrmcfg.sandboxiot.hillrom.com/apps/remotemanagement/index.html#/		
Description			
<p>The application uses single-factor authentication to authenticate privileged users to the system. Single-factor authentication refers to the use of a single component to identify an end user of an application or system. The factor provided may be something the user knows, something the user is, or something the user has. Each of these options provides its own set of advantages and risks when used for authentication:</p> <ul style="list-style-type: none">• "Something you know", such as a user-defined password, may be easily created and changed when necessary. Authentication factors derived from the end user must have some degree to be managed by the user themselves, leaving the known secret's security up to them. This can result in the secret being forgotten or exposed through a breach of a separate system that holds or uses the same known secret.• "Something you are", such as a fingerprint, provides an end user with a constant factor that cannot be easily acquired or mimicked by an attacker. While this initially provides a strong barrier to entry and will always be with the end user, a single breach could leave the attribute used for authentication useless as it cannot be updated.• "Something you have", such as a hardware token, can be managed from a central source and is configured to constantly update, removing responsibility for the known secret from the user. However, this transition of the knowledge base may hinder the application's accessibility if the device is not always at hand.			
Impact			
If an attacker compromises the authentication mechanism (e.g., a victim's account password), they will have full access to functionality and data normally only available to the victim.			
Recommendation			
<p>Multi-factor authentication should be implemented and enforced for externally accessible applications containing sensitive data or functionality. Multi-factor authentication is built upon the combination of two or more components that can prove a user's identity to the application. This provides an additional layer of security as it is assumed that an unauthorized attacker will not be able to supply both factors required for authentication. The factors required by the application should be a combination of at least two distinct factors from the following:</p> <ul style="list-style-type: none">• Something the user knows• Something the user is• Something the user has <p>For example, a common multi-factor authentication mechanism requires a user to provide a password they have created (something they know), as well as a value from a hardware token (something they have). If an attacker can compromise a user's password, they will still not have access to the hardware token and will not be able to gain access to the system.</p> <p>Note: requiring two or more pieces of information for authentication that fall under the same factor category does not provide true multi-factor authentication. For example, a user's password and the answer to their security question are both something the user knows. Requiring both during authentication does not represent true multi-factor authentication.</p>			

How to recreate the Security defect

- Browse application
- Perform any critical operation (like Firmware upload for administrator)
- Observe there is no additional authentication step for any critical operations.

3. Abbreviation

APP	Application
HTML	Hyper Text Mark-up Language
HTTP(S)	Hypertext transfer protocol (Secured)
Pg.	Page
TLS	Transport Layer Security
SSL	Secure Sockets Layer
IP	Internet Protocol
LTTS	Larsen & Toubro Technology Services
SOP	Same Origin Policy
OWASP	Open Web Application Security Project
VAPT	Vulnerability Assessment and Penetration testing
CSP	Content Security Policy
CORS	Cross Origin Resource Sharing
IDOR	Insecure direct object references
MFA	No Multifactor Authentication
URL	Uniform Resource locator
XSS	Cross-Site Scripting
XXE	XML External Entities
SQL	Structured Query Language
SSO	Single Sign-On
CSRF	Cross Site Request Forgery